



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **09034864 A**(43) Date of publication of application: **07.02.97**

(51) Int. Cl. **G06F 15/78**
G06F 11/22
G06F 11/28

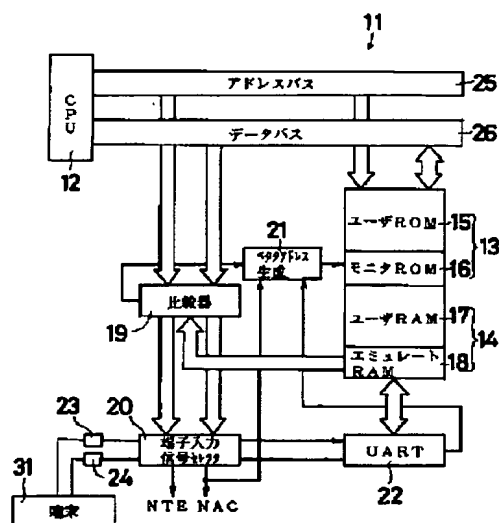
(21) Application number: **07178449**(71) Applicant: **SHARP CORP**(22) Date of filing: **14.07.95**(72) Inventor: **ONO KENJI**(54) **SINGLE CHIP MICROCOMPUTER**

(57) Abstract:

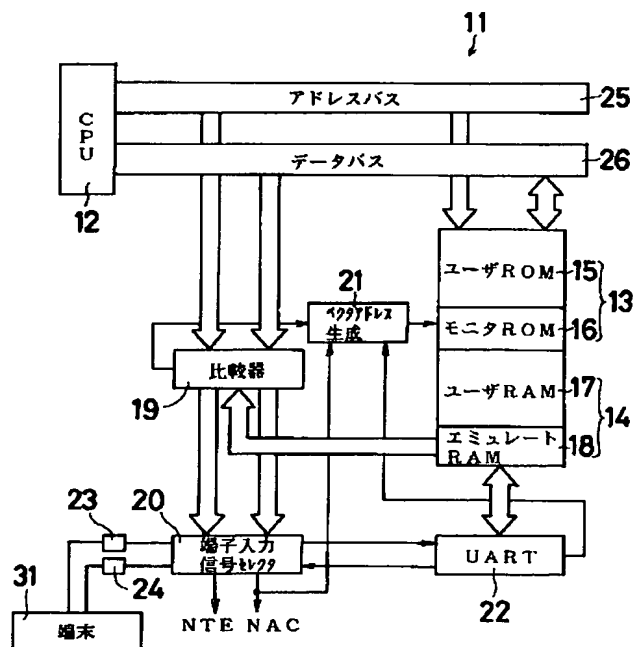
PROBLEM TO BE SOLVED: To provide the single chip microcomputer in which an application program is corrected while being mounted on a device without increasing the number of terminals of the single chip microcomputer.

SOLUTION: An ACL terminal 23 and a TEST terminal 24 in the single chip microcomputer 11 are part of external connection terminals prepared in advance for program operation. When a debug program stored in a monitor ROM 16 is started, a connection state in a terminal input signal selector 20 is switched to send/receive data required for debugging the user program to/ from a terminal equipment 31 via the ACL terminal 23 and the TEST terminal 24.

COPYRIGHT: (C)1997,JPO



(11)特許出願公開番号



【特許請求の範囲】

【請求項 1】 マイクロプロセッサと、
マイクロプロセッサによって実行されるアプリケーション
プログラムと、アプリケーションプログラムをデバッ
グするためのデバッグ用プログラムとを格納する ROM
と、
アプリケーションプログラムおよびデバッグ用プログラ
ムに関連する情報が一時的に格納される RAM と、
複数の外部接続端子とを備えるシングルチップマイクロ
コンピュータにおいて、
一部の外部接続端子を介して外部の装置と情報の授受を
行うことが可能な通信用のインタフェースと、
前記一部の外部接続端子の機能を切替えるように制御す
る制御手段とを備え、制御手段は、デバッグ用のプログラ
ムが起動されたときには前記一部の外部接続端子の接
続を通信用のインタフェースに切換え、アプリケーション
プログラムが起動されたときにはマイクロコンピュ
ータとしてのプログラム動作機能用に切換えることを特徴
とするシングルチップマイクロコンピュータ。

【請求項 2】 前記 ROM は、電気的に書換え可能な R
OM であることを特徴とする請求項 1 記載のシングルチ
ップマイクロコンピュータ。

【請求項 3】 前記一部の外部接続端子は、ACL 端子
または TEST 端子として機能することを特徴とする請
求項 1 または 2 記載のシングルチップマイクロコンピ
ュータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ROM (Read Onl
y Memory) に書込まれたプログラムを実行する機器組込
み型のシングルチップマイクロコンピュータに関する。

【0002】

【従来の技術】従来、一般的に用いられているシングル
チップマイクロコンピュータ（以下単にマイクロコンピ
ュータと称す。）は、機器に組込まれて使用され、内部
の ROM に格納されているユーザプログラムによって機
器の制御を行っている。一般的なマスク ROM 版マイク
ロコンピュータにおいては、マイクロコンピュータ内部
の情報を出力したり、ユーザプログラムの修正を行うた
めの専用の機能が無いために機器に組込まれた状態、す
なわち実際に動作する状態でユーザプログラムのデバッ
グを行うことが困難である。また、ユーザプログラムの
デバッグが可能であってもユーザプログラムはマイクロ
コンピュータの ROM に格納されているので、ROM が
マスク ROM である場合にはユーザプログラムの書換え
を行うことは困難である。

【0003】上述のようなマイクロコンピュータの開発
は、ICE (In Circuit Emulator) を用いて行われ
る。ICE によってマイクロコンピュータの開発を行う
際には、ユーザプログラムが格納された ROM などの代

わりにエバリュエーションチップ（以後、「エバーチッ
プ」と称する）を用いる。エバーチップは、マイクロコ
ンピュータにおける端子からの出力だけでなく、内部の
構成要素をそれぞれ接続する信号線における信号レベル
を取出すことができるような構造となっており設けられ
る端子の数が多い。したがって、前記 ROM と同じ様に
機器に組込むことができない。そのため、前記 ROM が
挿入されるソケットなど、エバーチップ、ICE をケー
ブルで接続しなければならない。ケーブルによって接続
するのでケーブルを介して入力されるノイズなどの影響
についても考慮しなければならない。

【0004】また、エバーチップは、マスク ROM 版マ
イクロコンピュータとは製造プロセスが異なるので、電
気的な特性などが異なる可能性があり、マイクロコンピ
ュータの最終的な評価に用いるには問題がある。

【0005】マイクロコンピュータを組込もうとする機
器に直接装着してプログラムの動作評価を行うことがで
きるシングルチップマイクロコンピュータとして OTP
(One Time Programmable ROM) 版がある。OTP 版
は、外見的には前記マスク ROM 版と同様であるので機
器に組込んでプログラムの動作評価を行うことができる
けれども、マイクロコンピュータ内部の情報を外部に出
力する機能を有していないので、プログラムのデバッグ
を行うには非常に手間がかかる。

【0006】上述のような従来の技術における問題点を
解決するための第 1 の技術は、特開平 4-332051
号公報に開示されている。第 1 の先行技術においては、
マイクロコンピュータに備えられる ROM にユーザプロ
グラムと共にデバッグ用のプログラムが格納されてい
る。マイクロコンピュータに設けられる通信用のインタ
フェースを介して端末装置に内部情報を出力し、端末装
置からデバッグコマンドなどが入力される。デバッグ開
始時には、ROM の内容を全てマイクロコンピュータ内
の RAM に読み込み、デバッグ用のプログラムは RAM に
対してアクセスするようにする。

【0007】また、第 2 の技術は、特開平 3-2029
24 号公報に開示されている。第 2 の先行技術におい
ては、デバッグ用のプログラムを記憶した装置とのインタ
フェースをマイクロコンピュータ内に設けている。マイ
クロコンピュータ内部の端子の切換え状態によってユー
ザプログラムを実行させるかユーザプログラムのデバッ
グを行うかを選択する。この先行技術では、デバッグ用
のプログラムはマイクロコンピュータ外部のデバッグ装
置に記憶されており、デバッグ装置のエミュレーション
用のメモリにユーザプログラムを読み込んでユーザプロ
グラムのデバッグを行う。機器に組込まれているマイク
ロコンピュータを介してデバッグ装置からのコマンドが入
力されて動作の確認が行われる。

【0008】また、第 3 の技術は特開昭 61-5805
2 号公報に開示されている。第 3 の先行技術におけるデ

バッグ装置では、アプリケーションプログラムが記憶されているROMに置換えて前記ROMが装着されるソケットにプローブを挿入してマイクロコンピュータと信号の授受を行う。デバッグ装置のRAMには、デバッグしようとするユーザプログラムが記憶されており、このプログラムに対してデバッグを行う。デバッグ用のプログラムはデバッグ装置内のROMに記憶されている。

【0009】

【発明が解決しようとする課題】第1の先行技術では、通信に用いている端子は、通信専用の端子でありマイクロコンピュータにおける端子数の増加を招いている。

【0010】また、第2の先行技術では、ユーザプログラムのデバッグを行うにはデバッグ用のプログラムを備えた装置が別途必要となり、容易にデバッグを行うことができない。データ転送用の端子が複数であるので端子の切換えを行う切換え装置が必要とする面積が大きくなる。

【0011】第3の先行技術では、ユーザプログラムなどを記憶しているROMに置換えてROMプローブを挿入してデバッグ装置とマイクロコンピュータとの間で信号の授受を行っている。プローブを介して信号の授受を行いながら実際にユーザプログラムが動作する速度でエミュレーションを行うことはタイミング的な仕様の限界から、次第に困難に成りつつある。また、プローブによる接続はマイクロコンピュータが取付けられる機器のたとえばROM用のソケットなどを破壊してしまう可能性があり安全性の点で問題がある。

【0012】本発明の目的は、シングルチップマイクロコンピュータにおける端子数を増加させることなく、機器に組込んだ状態でアプリケーションプログラムの修正を行うことができるシングルチップマイクロコンピュータを提供することである。

【0013】

【課題を解決するための手段】本発明は、マイクロプロセッサと、マイクロプロセッサによって実行されるアプリケーションプログラムと、アプリケーションプログラムをデバッグするためのデバッグ用プログラムとを格納するROMと、アプリケーションプログラムおよびデバッグ用プログラムに関連する情報が一時的に格納されるRAMと、複数の外部接続端子とを備えるシングルチップマイクロコンピュータにおいて、一部の外部接続端子を介して外部の装置と情報の授受を行うことが可能な通信のインタフェースと、前記一部の外部接続端子の機能を切換えるように制御する制御手段とを備え、制御手段は、デバッグ用のプログラムが起動されたときには前記一部の外部接続端子の接続を通信のインタフェースに切換え、アプリケーションプログラムが起動されたときにはマイクロコンピュータとしてのプログラム動作機能用に切換えることを特徴とするシングルチップマイクロコンピュータである。本発明に従えば、シングルチッ

プマイクロコンピュータは、マイクロプロセッサによって実行されるアプリケーションプログラムと、デバッグ用のプログラムとがROMに格納されている。アプリケーションプログラムのデバッグを行う際にはデバッグ用のプログラムを起動する。デバッグ用のプログラムが起動されると、制御手段は一部の外部接続端子の機能を切換える。機能が切換えられた外部接続端子を介して、通信のインタフェースは外部の装置との情報の授受を行う。デバッグ用プログラムは、外部の装置から供給される情報に基づいてアプリケーションプログラムのデバッグを行う。アプリケーションプログラムが起動されたときには、制御手段は外部接続端子をプログラム動作機能用に切換えて情報の授受を行う。したがって、アプリケーションプログラムのデバッグを行うために必要な外部の装置との情報の授受は、制御手段によって機能が切換えられた一部の外部接続端子を用いて行われるので、デバッグを行うために情報の授受専用の端子を設ける必要がなく、シングルチップマイクロコンピュータが形成されるチップの面積を広くしなくてもアプリケーションプログラムのデバッグを行うことができる。

【0014】また本発明の前記ROMは、電氣的に書換え可能なROMであることを特徴とする。本発明に従えば、ROMは電氣的に書換え可能なROM、たとえばフラッシュメモリなどによって構成される。したがって、デバッグ用プログラムによって修正されたアプリケーションプログラムをシングルチップマイクロコンピュータが機器に組込まれた状態のままでROMに記憶することができる。

【0015】また本発明の前記一部の外部接続端子は、ACL端子またはTEST端子として機能することの特徴とする。本発明に従えば、デバッグ用プログラムの動作時に外部の装置との情報の授受を行う一部の外部接続端子は、ACLまたはTEST端子である。したがって、シングルチップマイクロコンピュータとしての機能を行う際に使用頻度の低い外部接続端子の機能を切換えて外部の装置と情報の授受を行うので、アプリケーションプログラムのデバッグを行う際に、外部接続端子の機能を頻繁に切換える必要がなく制御手段を簡易な回路で実現することができる。

【0016】

【発明の実施の形態】図1は、シングルチップマイクロコンピュータ11の構成を示すブロック図である。シングルチップマイクロコンピュータ（以後単にマイクロコンピュータと称する）11は、CPU（中央処理装置）12と、ROM（Read Only Memory）13と、RAM（Random Access Memory）14と、比較器19と、端子入力信号セクタ20と、ベクタアドレス生成回路21と、UART（Universal Asynchronous Receiver and Transmitter）22とを含んで構成され、図示しない複数の外部接続用の端子を備える。図1のマイクロコンピュ

ータ11においては、一部の外部接続用端子としてACL (All Clear) 端子23とTEST端子24とを示した。

【0017】ROM13は、ユーザROM15とモニタROM16とによって構成され、たとえばフラッシュメモリなどの電氣的に書換えが可能なROMである。RAM14は、ユーザRAM17とエミュレートRAM18とによって構成される。CPU12には、アドレスバス25とデータバス26とが接続されている。ROM13とRAM14とは、CPU12とアドレスバス25とデータバス26とによって接続される。アドレスバス25によってROM13などのアドレスが指定され、データバス26を介してデータの授受が行われる。

【0018】ユーザROM15には、ユーザが開発したアプリケーションプログラムであるユーザプログラムが格納されており、モニタROM16には前記ユーザプログラムをデバッグするためのプログラムなどが記憶されている。デバッグ用のプログラムによって、外部の端末装置31との通信制御およびユーザROM15に格納されているユーザプログラムのスタート、ストップなどの制御を行う。

【0019】ユーザRAM17は、ユーザROM15に格納されているユーザプログラムの動作時にデータの一時書込みなどが行われる。エミュレートRAM18は、モニタROM16に格納されているデバッグ用プログラムの起動時に端末装置31から入力されたデータなどを格納する。

【0020】UART22は、モニタROM16に予め記憶されているデータの転送速度および一度に転送されるデータのビット数などの情報に基づいて、2本の信号線を用いてシリアル方式でデータの送受信を行う。UART22には、端子入力信号セクタ20を介して端末装置31からデータが入力される。UART22に入力されたデータは、エミュレートRAM18およびベクタアドレス生成回路21に入力される。

【0021】マイクロコンピュータ11は、ACL端子23に予め定めるハイレベルかローレベルかの信号が入力されることによって電源投入時と同様の動作を行い、マイクロコンピュータ11内のレジスタなどの値をクリアする。また、マイクロコンピュータ11は、電源投入時にTEST端子24に入力されている信号のレベルが、予め定めるレベルであったときには、テストモードと称される処理としてマイクロコンピュータ11内の回路が正常に動作しているかどうかを予め定められるプログラムに従って信号を入力することによって調べる。

【0022】図2は、UART22におけるデータのフォーマットを示す図である。図2(1)では、データ部分が7もしくは8ビットによって構成されたフォーマットの例を示す。7/8ビットモードにおいては、予め定めるスタートビットに引続いてビット0からビットn

(nは7もしくは8)までのデータによって構成されたデータ部分が設けられる。nが7である場合には、データ部分に引続いて1ビットのデータであるパリティビットが設けられるが、nが8である場合にはパリティビットは設けられない。データ部分もしくはパリティビットに引続いて1もしくは2ビットからなるストップビットが設けられる。

【0023】図2(2)では、データ部分が9ビットによって構成されたフォーマットの例を示す。9ビットモードにおいては、予め定めるスタートビットに引続いて、ビット0からビットnまでのデータとパリティビットとによって構成されたデータ部分が設けられる。データ部分に引続いては1もしくは2ビットからなるストップビットが設けられる。

【0024】比較器19は、アドレスバス25およびデータバス26のデータと、エミュレートRAM18に記憶されているデータとの比較を行う。ベクタアドレス生成回路21は、ACL信号の入力および各種割込み発生時、たとえば比較器19において行われた比較の結果などに基づいて、ユーザROM15およびモニタROM16における所定のアドレスを出力する。ベクタアドレス生成回路21には、たとえばモニタROM16におけるデバッグ用プログラムのスタートアドレスなどが格納されていて、ベクタアドレス生成回路21から出力されるアドレスに基づいてプログラムが実行される。

【0025】図3は、端子入力信号セクタ20の構成を示すブロック図である。端子入力信号セクタ20は、レジスタ41と、オートリセット回路42と、信号切換え回路43と、信号入力許可回路44と、入出力切換え回路45と、トランジスタ46とを含んで構成される。

【0026】入出力切換え回路45は、バッファ51と、n型のFET (Field Effect Transistor) であるトランジスタ52と、NANDゲート53と、p型のFETであるトランジスタ54とを含んで構成される。信号切換え回路43は、ANDゲート55、56とORゲート57とを含んで構成されており、ANDゲート55の出力がORゲート57の一方の入力端子に入力され、ANDゲート56の出力がORゲート57の他方の入力端子に入力される。信号入力許可回路44は、ANDゲート58を含んで構成される。

【0027】図3には1つのDフリップフロップとして示しているが、レジスタ41は端子入力信号セクタ20において複数個設けられており、各レジスタ41を識別するためのアドレスがそれぞれに割付けられている。アドレスバス25を介して与えられるアドレス値によって指定されるレジスタ41に内部クロック信号に基づいてデータバス26のデータが書込まれる。内部クロック信号がレジスタ41の入力端子CKに入力されると入力端子Dに入力されている信号のレベルが出力端子Qから

出力され、反転出力端子QBからは出力端子Qの出力を反転した信号が出力される。出力端子Qから出力される信号SEL1は、ANDゲート58の一方の入力端子に入力される。出力端子QBから出力される信号SEBは、ANDゲート55、56の一方の入力端子にそれぞれ入力され、かつバッファ51の制御端子に入力される。

【0028】オトリセット回路42の出力は、レジスタ41のリセット端子Rと、ANDゲート55の他方の入力端子に入力される。ACL端子23に接続される信号線60は、ANDゲート56の他方の入力端子とANDゲート58の他方の入力端子とに接続される。

【0029】入出力切換え回路45におけるバッファ51には、TEST端子24から信号線59を介して信号が入力される。信号線59には、p型のFETであるトランジスタ46のドレインDが接続されている。トランジスタ46は、ゲートGが接地されており、ソースSには所定の電位が与えられている。したがって、TEST端子24に対して外部からの入力がない場合は、入出力切換え回路45に入力される信号のレベルはハイレベルとなる。

【0030】バッファ51には、信号SEBが制御信号として入力されており、信号SEBがローレベルとなるとバッファ51の信号出力がハイインピーダンス状態となる。バッファ51の出力は、内部TEST信号NTEとしてマイクロコンピュータ11内の他の構成要素に供給される。なお、バッファ51の出力にはトランジスタ54のドレインDが接続されている。このトランジスタ54は、ゲートGが接地されており、ソースSには所定の電位が与えられている。したがって、バッファ51の信号出力がハイインピーダンス状態の場合、内部TEST信号NTEはハイレベルとなる。

【0031】トランジスタ54のドレインDは、さらにNANDゲート53の第1の入力端子に接続される。NANDゲート53の第2の入力端子には信号SEL1が入力され、第3の入力端子にはUART22からの出力信号OUAが入力される。NANDゲート53の出力は、トランジスタ52のゲートGに与えられる。トランジスタ52は、ドレインDが接地され、ソースSが信号線59に接続される。

【0032】マイクロコンピュータ11に電源が投入されてからの動作について説明する。電源が投入されると、オトリセット回路42が正のパルス信号を出力し、レジスタ41がリセットされる。したがって、信号SEL1はローレベルとなり、信号SEBはハイレベルとなる。信号SEBがバッファ51の制御端子に入力されているので、TEST端子24に入力されている信号のレベルがそのまま内部TEST信号NTEのレベルとして出力される。内部TEST信号NTEは、マイクロコンピュータ11をTESTモードで動作させる時のみ

ローレベルとなる。

【0033】アドレスバス25によって示されるレジスタ41は、入力端子Dに入力されるデータバス26の信号値を内部クロック信号CKに同期してラッチする。入力端子Dに入力される信号がハイレベルである場合、信号SEL1がハイレベルになる。信号SEL1がハイレベルになることによって、ACL端子23に入力される信号が、信号入力許可回路44のANDゲート58を介してUART入力信号UAとしてUART22に入力される。信号入力許可回路44の出力がUART22に供給されることによって、モニタROM16のデバッグ用プログラムが起動する。デバッグ用プログラムは、ユーザROM15に格納されているアプリケーションプログラムのスタートアドレスの値をエミュレートRAM18に記憶させる。また、デバッグ用プログラムは、予め定めるデータをUART22に供給する。

【0034】UART22は、前記予め定めるデータをUART出力信号OUAとして端子入力信号セクタ20の入出力切換え回路45に入力する。入力されたデータの信号レベルによってNANDゲート53の出力が定まる。NANDゲート53の出力がトランジスタ52のゲートGに入力されているので、前記データの信号レベルがハイレベルである場合にはトランジスタ52が導通し、TEST端子24の出力がローレベルとなる。また、前記データの信号レベルがローレベルである場合にはトランジスタ52は遮断し、TEST端子24からの出力信号がハイレベルとなる。

【0035】端末装置31がACL端子23およびTEST端子24に接続されている場合は、TEST端子24を介して入力されるデータに基づいて端末装置31はACL端子23およびTEST端子24に接続されていることを示す応答信号であるデータをACL端子23に供給する。

【0036】マイクロコンピュータ11においては、TEST端子24から前記データを出力してから予め定める時間以内にACL端子23に端末装置31から応答信号が入力されていない場合には、端末装置31が接続されていないと判断する。すなわち、端末装置31から応答信号が入力されない場合には、マイクロコンピュータ11はプログラムデバッグモードではないと判断する。

【0037】端末装置31から応答信号が入力されない、デバッグ用プログラムはレジスタ41にデータバス26を介して信号を入力することによってレジスタ41をクリアしてACL端子23への入力を内部ACL信号NACとし、TEST端子24への入力が内部TEST信号NTEとなるようにする。ACL端子23、TEST端子24の機能切換え後、エミュレートRAM18に記憶させていたユーザプログラムのスタートアドレスをアドレスバス25に戻してユーザプログラムを動作させる。

【0038】端末装置31から応答信号が入力されると、前記応答信号はACL端子23、信号入力許可回路44を介してUART22に入力される。UART22に入力された応答信号に基づいてACL端子23およびTEST端子24が通信専用端子として使用される。デバッグ用プログラムは、プログラムデバッグモードであることが確認されたので、引続いてACL端子23を介して入力されるデバッグコマンドを待ち受ける。

【0039】プログラムデバッグモードにおける処理について説明する。端末装置31において入力されたデバッグコマンドは、ACL端子23を介して端子入力信号セクタ20へと入力される。さらにUART22に入力され、UART22からエミュレートRAM18に格納されデバッグ用プログラムによって解析される。

【0040】解析されたコマンドがレジスタなどの内部データを参照するコマンドである場合は、デバッグ用プログラムはアドレスバス25およびデータバス26から必要な情報を取り込みUART22へと出力する。UART22は、与えられた情報をTEST端子24を介して端末装置31へと送信する。

【0041】解析されたコマンドがブレイクポイントデータの設定である場合は、動作中のデバッグ用プログラムは端末装置31から送られてきたデータをエミュレートRAM18に格納する。解析されたコマンドがユーザプログラムの実行を指示するコマンドである場合は、エミュレートRAM18に格納されているユーザプログラムのスタートアドレスをアドレスバス25に設定し、示されたアドレスからユーザプログラムを実行させる。プログラムデバッグモードにおいてユーザプログラムを実行する際に前もってブレイクポイントデータを設定するとブレイクポイントデータと各バス25、26のデータとを比較器19において比較する。

【0042】エミュレートRAM18に格納されているブレイクポイントデータがアドレス値である場合には、このアドレス値とアドレスバス25を介して転送されるアドレス値とが一致したときにユーザプログラムを停止させる。エミュレートRAM18に格納されているブレイクポイントデータがアドレス値以外の所定のデータである場合には、このデータとデータバス26を介して転送されるデータとが一致したときにユーザプログラムを停止させる。ユーザプログラムが停止すると停止したユーザプログラムのアドレスがエミュレートRAM18に格納されモニタROM16のデバッグ用プログラムが再び起動される。

【0043】図4は、マイクロコンピュータ11における各信号のタイミングチャートである。時刻t0において、図4(1)に示す電源が投入されると、時刻t1において図4(2)に示すオトリセット回路42の出力がロー「L」レベルからハイ「H」レベルへと立上がる。オトリセット回路42の出力が、ハイレベルとな

ったことによって図4(3)に示す内部ACL信号NACが立上がる。また、図4(9)に示すTEST端子24への入力が立上がりはじめ、図4(7)に示す内部TEST信号NTEが立上がりはじめる。

【0044】オトリセット回路42の出力が、時刻t2においてローレベルへと立下がると、内部ACL信号NACがローレベルと立下がる。オトリセット回路42の出力がローレベルになったことによって、図4

(6)に示す信号SEL1が立上がりはじめ時刻t3においてハイレベルとなる。信号SEL1がハイレベルになったことによって、時刻t3から図4(8)に示すUART出力信号OUAとして前述した所定のデータが端末装置31へとTEST端子24を介して出力される。なお、TEST端子24から出力されるデータは図4

(9)に示すように信号OUAとはハイレベルとローレベルとが入替わる。前記データの送出は、時刻t3から時刻t4までの期間T1において行われる。端末装置31は、前記データが入力されると端末装置31が接続されていることを示す応答信号をマイクロコンピュータ11へと送出する。

【0045】図4(4)に示す端末装置31から送出された応答信号が、時刻t5においてACL端子23に入力されると、図4(5)に示すUART入力信号UAとしてUART22へと入力される。応答信号の受取りは時刻t5から時刻t6までの期間T2において行われる。期間T2においてACL端子23に入力される信号とUART入力信号UAとは同一レベルとなる。オトリセット回路42の出力が立下がる時刻t2から応答信号の受取りが完了する時刻t6までの期間T3において、デバッグ用プログラムの起動およびUART22に対する応答信号の入力が待ち受けられる。期間T2において応答信号が入力されると時刻t6以降はプログラムデバッグモードとなる。

【0046】図5は、マイクロコンピュータ11における処理を示すフローチャートである。ステップs1では、電源が投入されるのを待ち受ける。電源が投入されるとステップs2に進む。ステップs2では、オトリセット回路42において信号が発生され、レジスタ41がリセットされる。

【0047】続くステップs3では、内部TEST信号NTEの信号レベルがハイレベルであるかどうかを判断する。内部TEST信号NTEの信号レベルがハイレベルであるときには、ステップs4に進む。ステップs4では、オトリセット回路42からの信号出力が停止され、レジスタ41のリセット状態が解除される。リセット状態が解除されたので、レジスタ41はデータバス26の値を内部クロック信号に基づいて取込む。したがって、レジスタ41の出力端子Qから出力される信号SEL1がハイレベルとなる。

【0048】信号SEL1がハイレベルになることによ

って、信号切換え回路43から出力される内部ACL信号NACの信号レベルがローレベルとなる。また、信号入力許可回路44から出力されるUART入力信号UAの信号レベルは、ACL端子23から入力される信号のレベルに基づいて定められる。すなわち、ACL端子23は端末装置31から送られてくるデータ信号の入力端子となる。

【0049】ステップs5では、ベクタアドレス生成回路21によって指定されるアドレスに基づいてモニタROM16に格納されているデバッグ用プログラムが起動される。また、ユーザROM15に格納されているユーザプログラムのスタートアドレスがエミュレートRAM18に格納される。

【0050】続くステップs6においては、UART22から所定の信号が出力されTEST端子24を介して端末装置31へと与えられる。TEST端子24は端末装置31へとデータ信号を出力する出力端子となる。ステップs7では、前記所定の信号に対する端末装置31からの応答信号がACL端子23を介して入力されたかどうかを判断する。応答信号を受信した場合は前述したプログラムデバッグ処理を行う。

【0051】また、応答信号を受信しなかった場合には、端末装置31が接続されていないと判断し、信号SEL1をローレベルにする。信号SEL1がローレベルになることによって、ACL端子23に入力される信号は、マイクロコンピュータ11において内部ACL信号NACとして用いられ、TEST端子24に入力される信号は内部TEST信号NTEとして用いられる。さらに、エミュレートRAM18に格納されているユーザプログラムのスタートアドレスを読み出しユーザプログラムを起動させる。

【0052】ステップs3において、TEST端子23に入力されている信号がローレベルであると判断した時には、マイクロコンピュータ11は前述したテストモードに入る。

【0053】以上のように本発明の実施のこの形態によれば、マイクロコンピュータ11を起動する際に、ACL端子23とTEST端子24とに端末装置31が接続されているかどうかを確認する。端末装置31が接続されている場合には、端子入力信号セクタ20において、端子の機能を切換えてACL端子23、TEST端子24を通信専用の端子とする。ユーザプログラムをデバッグするプログラムはモニタROM16に格納されているので、端末装置31は送られてきたデータを表示したり、デバッグコマンドを入力することができるような構成であればよく、プログラムデバッグのための専用の装置を用いる必要がなく、容易にプログラムのデバッグを行うことができる。

【0054】また、ACL端子23およびTEST端子24の機能は、通常のプログラム動作機能時にはあまり

使用されない端子であるので、端子入力信号セクタ20の構造を簡易な構造とすることができる。

【0055】なお、本発明の実施のこの形態では、ROM13は書換え可能なROMであるとしたが、マスクROMのような書換えることができないROMを使用した場合であっても、デバッグ情報を端子入力信号セクタ20を介して読み出すことができるので、マイクロコンピュータ11が実際に機器に取付けられた状態でデバッグを行うことができる。

10 【0056】

【発明の効果】以上のように本発明によれば、アプリケーションプログラムのデバッグを行うために必要な外部の装置との情報の授受は制御手段によって機能が切換えられた一部の接続端子を用いて行われるので、通信専用の端子をシングルチップマイクロコンピュータに設ける必要がなく、シングルチップマイクロコンピュータが形成されるチップの面積を広くすることなく機器に取付けられた状態でROMに格納されているアプリケーションプログラムのデバッグを行うことができる。

20 【0057】また本発明によれば、ROMは電氣的に書換え可能なROMによって構成されるので、シングルチップマイクロコンピュータを機器に組込んだ状態のままデバッグ用プログラムによってアプリケーションプログラムを修正することができるので、アプリケーションプログラムのデバッグを容易に行うことができ、アプリケーションプログラムの開発にかかる時間を短縮することができる。また、デバッグ用プログラムの書換えが可能となるので、ユーザ独自の制御を行うことができる。

30 【0058】さらに本発明によれば、マイクロコンピュータとしての機能を行う際に使用頻度の低い外部接続端子の機能を切換えて外部の装置と情報の授受を行うので、アプリケーションプログラムのデバッグを行う際に外部接続端子の機能を頻繁に切換える必要がなく制御手段を簡易な回路で実現することができる。

【図面の簡単な説明】

【図1】本発明の実施の一形態であるシングルチップマイクロコンピュータ11の構成を示すブロック図である。

40 【図2】UART22におけるデータのフォーマットを示す図である。

【図3】端子入力信号セクタ20の構成を示すブロック図である。

【図4】マイクロコンピュータ11における各信号のタイミングチャートである。

【図5】マイクロコンピュータ11における処理を示すフローチャートである。

【符号の説明】

11 シングルチップマイクロコンピュータ

12 CPU

50 13 ROM

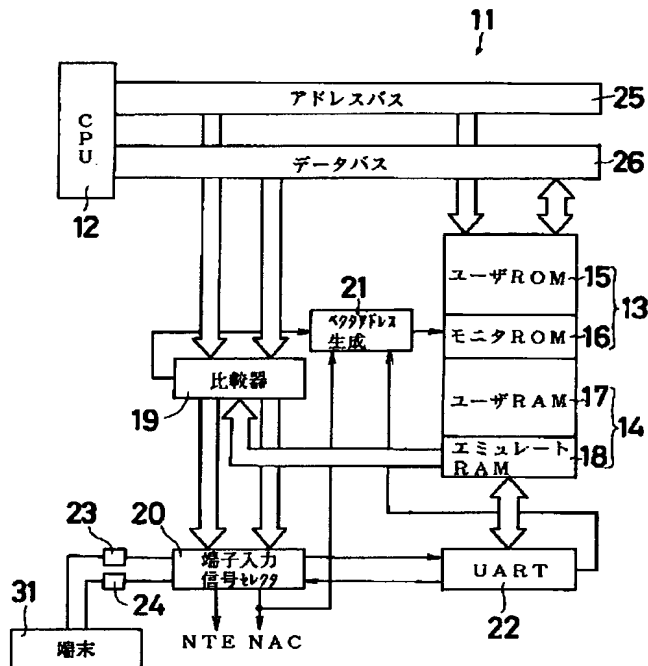
13

14

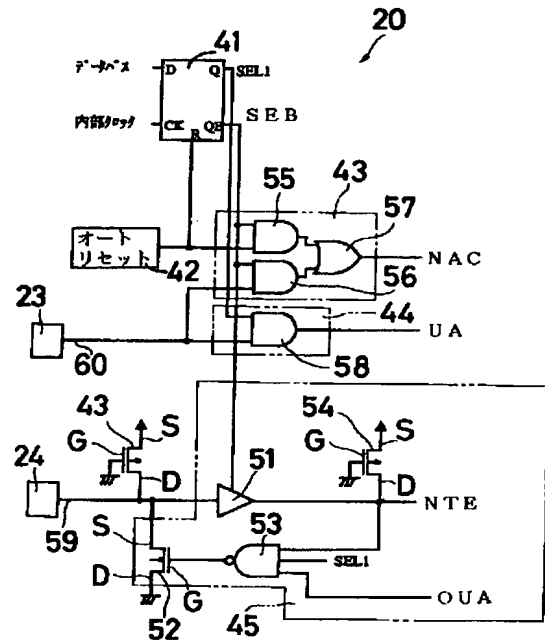
- 14 RAM
 15 ユーザROM
 16 モニタROM
 17 ユーザRAM
 18 エミュレートRAM
 19 比較器
 20 端子入力信号セクタ

- * 21 ベクタアドレス生成回路
 22 UART
 23 ACL端子
 24 TEST端子
 25 アドレスバス
 26 データバス
 * 31 端末装置

【図1】

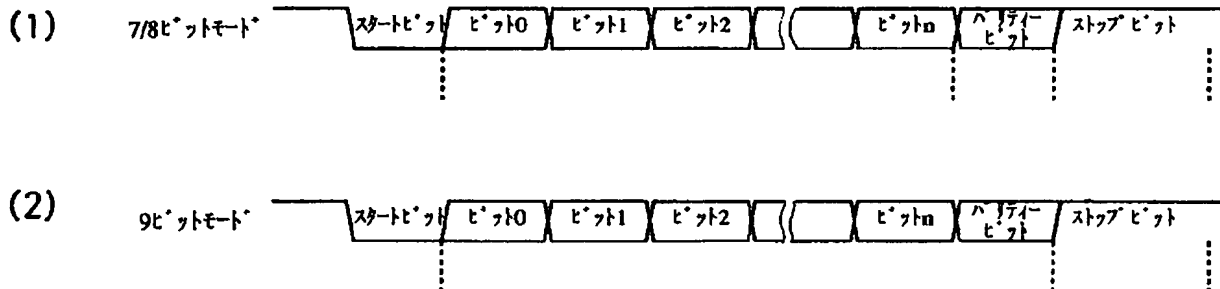


【図3】

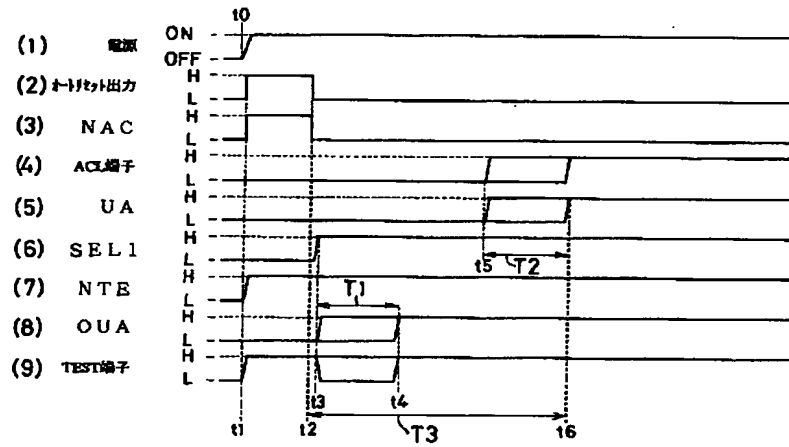


【図2】

UART送受信フォーマット



【図4】



【図5】

